

PROTOYPE GERBANG LOGIKA (AND, OR, NOT, NAND, NOR) PADA LABORATORIUM ELEKTRONIKA STMIK ROYAL KISARAN

Helmi Fauzi Siregar ^{*1}, Ikhsan Parinduri ²

^{*1} Program Studi Teknik Informatika, Universitas Asahan,

² Program Studi Sistem Komputer, STMIK Royal Kisaran,

^{*1} Jl. Jend. Ahmad Yani Kisaran 21244, Sumatera Utara,

Telp : (0623) 347222

² Jl. Prof. H.M. Yamin 173 Kisaran 21222, Sumatera Utara,

Telp : (0623) 41079

fauzi.helmi.hf@gmail.com^{*1}, ikhsanparinduri9@gmail.com²

Abstrak - Prototype gerbang logika bertujuan untuk memenuhi kebutuhan dan kelancaran proses belajar mengajar pada salah satu materi perkuliahan rangkaian digital. Pembuktian dari logika gerbang OR, AND, NOT, NOR, dan NAND. Prinsip kerja prototype gerbang logika adalah berkerja berdasarkan logika input diantaranya adalah 0 dan 1. Untuk gerbang logika AND adalah gerbang perkalian input terdiri dari (0,0, 0,1, 1,0, 1,1) dan output terdiri dari 1 untuk high (1) dan 3 untuk low (0). Untuk gerbang OR adalah gerbang penjumlahan input terdiri dari (0,0, 0,1, 1,0, 1,1) dan output terdiri dari 3 high (1) dan 1 low (0). Untuk gerbang NAND adalah gerbang pembalik logika dari gerbang AND input terdiri dari (0,0, 0,1, 1,0, 1,1) dan output terdiri dari 3 high (1) dan 1 low (0). Untuk gerbang logika NOR input terdiri dari (0,0, 0,1, 1,0, 1,1) dan output terdiri dari 1 untuk high (1) dan 3 untuk low (0). Untuk gerbang NOT adalah gerbang invers dengan input (1, 0) dan output terdiri dari (0,1).

Kata Kunci - Gerbang Logika, Prototype, OR, AND, NOT, NOR, NAND.

Abstract - Prototype logic gate aims to meet the needs and fluency of teaching and learning process in one of the lecture material of digital circuit. Proof of gate logic OR, AND, NOT, NOR, and NAND. The working principle of the logic gate prototype is to work on the logical inputs of which are 0 and 1. For AND logic gates is the multiplicity of input multiplying (0,0, 0,1, 1,0, 1,1) and the output consists of 1 for high (1) and 3 for low (0). For the OR gate is the input summing gate composed of (0, 0,1, 1,0, 1,1) and the output consists of 3 high (1) and 1 low (0). For the NAND gate the inverting gate of the logic of the AND gate input consists of (0,0, 0,1, 1,0, 1,1) and the output consists of 3 high (1) and 1 low (0). For logic gates NOR inputs consist of (0,0, 0,1, 1,0, 1,1) and the output consists of 1 for high (1) and 3 for low (0). For NOT gate is the inverse gate with input (1, 0) and the output consists of (0,1).

Keywords - Logic Gate, Prototype, OR, AND, NOT, NOR, NAND.

I. PENDAHULUAN

A. Latar Belakang Masalah

Gerbang Logika dasar merupakan dasar pembentukan sistem digital. Gerbang logika

beroperasi dengan bilangan biner, sehingga disebut juga gerbang logika biner yang sering ditemukan dalam sirkuit digital yang diimplementasikan secara elektronik dengan menggunakan diode atau transistor.

Gerbang logika terdiri dari gerbang dasar :

1. Gerbang *AND* (Gerbang perkalian)
2. Gerbang *OR* (Gerbang penjumlahan)
3. Gerbang *NOT* (Gerbang *Invers* atau kebalikan)
4. Gerbang *NAND* (Gerbang *invers* dari gerbang *AND*)
5. Gerbang *NOR* (Gerbang *invers* dari gerbang *NOR*).

Seiring dengan perkembangan dunia digital yang sangat canggih, maka penulis membuat suatu penelitian guna untuk pembelajaran kepada mahasiswa agar mengetahui bagaimana cara kerja sistem digital menghasilkan logika biner nol (0) atau satu (1).

Banyak sekarang ini orang khususnya mahasiswa di jurusan ilmu komputer tidak mengetahui bagaimana sistem digital seperti komputer menghasilkan instruksi yang berlogika biner nol (0) atau satu (1). Maka dari penulis membuat suatu alat untuk memudahkan mahasiswa di dalam memahami sistem digital yang nantinya dapat bisa digunakan dalam praktek. Dengan demikian maka penulis membuat suatu penelitian dan mengajukan judul **“Prototype Gerbang Logika (*AND*, *NOT*, *NAND*, *NOR*) Pada Laboratorium Elektronika STMIK Royal Kisaran”**.

B. Perumusan Masalah

Berikut ini rumusan masalah pada penelitian ini adalah :

1. Bagaimana rancangan *prototype* gerbang logika sebagai media pembelajaran bagi mahasiswa?
2. Bagaimana ke-5 (lima) gerbang logika yang digunakan menghasilkan bilangan yang bernilai logika “0” atau “1” ?
3. Bagaimana sistem gerbang logika yang dirancang menghasilkan informasi bagi user ketika sistem mendeteksi tegangan?

C. Batasan Masalah

Batasan masalah yang diambil adalah :

1. Model *prototype* sistem gerbang logika yang dirancang menggunakan UML (*Unified Modelling Language*)
2. Sistem gerbang logika yang dibuat pada penelitian ini menggunakan 5 (lima) IC gerbang logika (*AND*, *OR*, *NOT*, *NAND*, *NOR*)
3. Tampilan indikator gerbang logika hanya menggunakan lampu LED pada masing-masing gerbang logika
4. Sumber tegangan dari baterai 9 V sebagai *power supply*
5. Untuk pemberian nilai logika yang bernilai “0” dan “1” diberikan melalui tombol *push button* yang ditekan oleh *user*.

D. Manfaat Penelitian

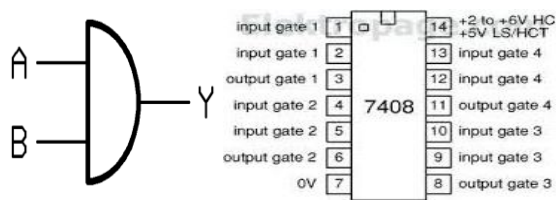
Manfaat penelitian ini adalah:

1. Membantu mahasiswa mempelajari gerbang logika dasar pada mata kuliah rangkaian digital dan sistem digital
2. Pembuatan media pembelajaran praktek rangkaian digital dan sistem digital.

II. LANDASAN TEORI

A. Gerbang *AND*

Gerbang *AND* digunakan untuk menghasilkan logika 1 jika semua masukan berlogika 1, jika tidak maka *output* yang dihasilkan akan berlogika 0 [5]. Simbol dan tabel kebenaran dari gerbang logika *AND* dapat dilihat berikut ini.



Gambar 2.1 Simbol Gerbang Logika AND

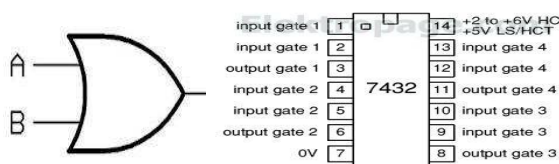
Tabel 2.1. Tabel Kebenaran dari Gerbang AND

INPUT		OUTPUT
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Dari tabel kebenaran pada tabel 2.1 di atas, dapat diketahui bahwa *output* pada gerbang AND hanya akan berlogika 1 jika kedua inputnya berlogika 1.

B. Gerbang OR

Gerbang OR digunakan untuk menghasilkan logika 0 jika semua masukan berlogika 0, jika tidak maka *output* yang dihasilkan akan berlogika 1 [5]. Simbol dan tabel kebenaran dari gerbang logika OR dapat dilihat berikut ini.



Gambar 2.2 Simbol Gerbang Logika OR

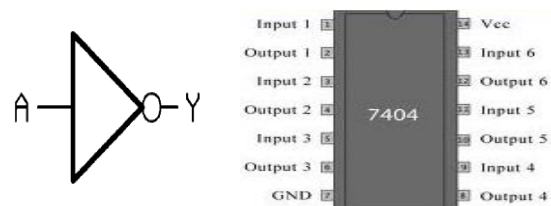
Tabel 2.2. Tabel Kebenaran dari Gerbang OR

INPUT		OUTPUT
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Dari tabel kebenaran pada tabel 2.2 di atas, dapat diketahui bahwa *output* pada gerbang OR hanya akan berlogika 0 jika kedua inputnya berlogika 0 dan jika tidak akan berlogika 1.

C. Gerbang NOT

Gerbang NOT digunakan untuk menghasilkan logika yang kebalikan dari inputnya [5].



Gambar 2.3 Simbol Gerbang Logika NOT

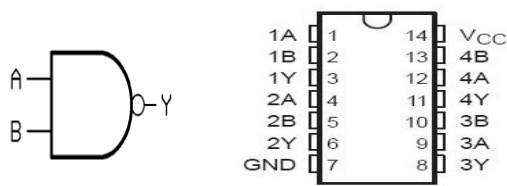
Tabel 2.3 Tabel Kebenaran dari Gerbang NOT

INPUT		OUTPUT
A		Y
0		1
1		0

Dari tabel kebenaran pada tabel 2.3 di atas, dapat diketahui bahwa *input* pada gerbang NOT menghasilkan *output* yang berlawanan dengan logika inputnya.

D. Gerbang NAND

Gerbang NAND merupakan gerbang AND yang outputnya gerbang NOT, digunakan untuk menghasilkan logika 0 jika semua masukan berlogika 1, jika tidak maka *output* yang dihasilkan akan berlogika 1 [5]. Simbol dan tabel kebenaran dari gerbang logika NAND dapat dilihat berikut ini.



Gambar 2.4 Simbol Gerbang Logika NAND

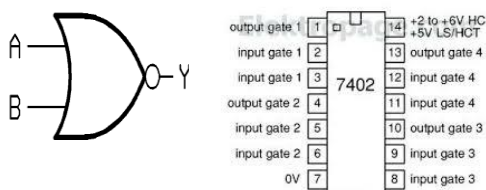
Tabel 2.4 Tabel Kebenaran dari Gerbang NAND

INPUT		OUTPUT
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Dari tabel kebenaran pada tabel 2.4 di atas, dapat diketahui bahwa *output* pada gerbang NAND hanya akan berlogika 0 jika kedua inputnya berlogika 1.

E. Gerbang NOR

Gerbang NOR merupakan gerbang OR yang outputnya ditambah dengan gerbang NOT, digunakan untuk menghasilkan logika 1 jika semua masukan berlogika 1, jika tidak maka *output* yang dihasilkan akan berlogika 0 [5]. Simbol dan tabel kebenaran dari gerbang logika AND dapat dilihat dibawah ini.



Gambar 2.5 Simbol Gerbang Logika NOR

Tabel 2.5 Tabel Kebenaran dari Gerbang NOR

INPUT		OUTPUT
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Dari tabel kebenaran pada tabel 2.5 di atas, dapat diketahui bahwa *output* pada gerbang NOR hanya akan berlogika 1 jika kedua inputnya berlogika 0.

F. UML (*Unified Modelling Language*)

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak [1]. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Untuk menguasai UML, sebenarnya cukup dua hal yang harus kita perhatikan :

1. Menguasai pembuatan diagram UML
2. Menguasai langkah-langkah dalam analisa dan pengembangan dengan UML

UML menggunakan notasi grafis untuk menyatakan suatu desain. Pemodelan dengan UML berarti menggambarkan yang ada dalam dunia nyata kedalam bentuk yang dapat dipahami dengan menggunakan notasi standart UML. Pemodelan dengan UML terdiri dari 8 tipe diagram yang berbeda untuk memodelkan sistem perangkat lunak. Masing-masing diagram UML didesain untuk menunjukkan satu sisi dari bermacam-macam sudut pandang (perspektif) dan terdiri dari tingkat abstraksi yang berbeda. Ke-8 (delapan) model tersebut adalah :

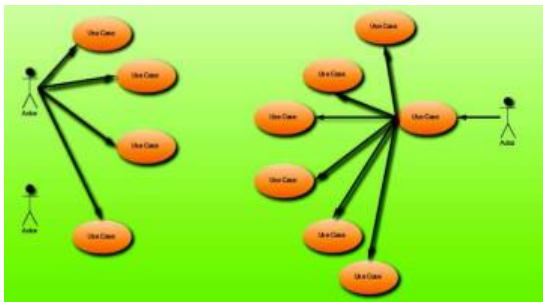
1. *Use case Diagram*
2. *Class Diagram*
3. *Object Diagram*
4. *State Diagram*
5. *Activity Diagram*
6. *Sequence Diagram*

7. *Collaboration Diagram*
8. *Component Diagram*
9. *Deployment Diagram*

Abstraksi konsep dasar UML yang terdiri dari *structural classification*, *dynamic behavior*, dan *model management*, bisa kita pahami dengan mudah apabila kita melihat gambar diatas dari diagram. *Main concepts* bisa kita pandang sebagai term yang akan muncul pada saat kita membuat diagram. Dan *view* adalah kategori dari diagram tersebut [1].

1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana” membuat sistem. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. Berikut ini adalah gambar 2.6 dari contoh *Use Case Diagram* :



Gambar 2.6 Contoh *Use Case Diagram*

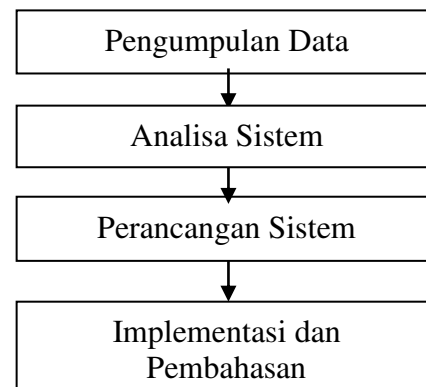
2. State Machine Diagram

State Machine Diagram adalah teknik yang umum digunakan untuk menggambarkan *behavior* sebuah sistem. Bentuk *state* diagram berfungsi untuk pemodelan suatu *class* yang perilakunya tergantung

pada *state* tertentu. Suatu *class* mempunyai diagram *state* sendiri jika mempunyai perilaku dinamis yang menarik. Gambar 2.5 menunjukkan contoh sebuah model *State Machine Diagram* [7].

III. METODOLOGI PENELITIAN

A. Kerangka Kerja Penelitian



1 Pengumpulan Data

Pengumpulan data merupakan hal yang sangat penting dalam sebuah penelitian. Dalam penelitian ini pengumpulan data penulis lakukan melalui :

a. Jurnal

Jurnal-jurnal yang penulis jadikan sebagai referensi adalah jurnal yang berkaitan dengan sistem gerbang logika yang berhubungan dengan judul yang penulis tulis.

b. Buku yang berhubungan dengan penelitian yang dilakukan

Buku yang penulis gunakan sebagai referensi adalah buku yang berkaitan dengan judul yang penulis tulis.

2 Analisa Sistem

Metode analisa sistem yang digunakan adalah Metode *Deskriptif*. Pada metode ini data yang ada dikumpulkan, disusun, dikelompokkan, dianalisa

sehingga diperoleh beberapa gambaran yang jelas pada masalah penelitian tersebut.

3 Perancangan Sistem

Berikut ini tahapan-tahapan dalam perancangan yang penulis lakukan adalah :

- a. Mempelajari Literatur
Membaca berbagai sumber terkait dengan permasalahan yang akan ditemukan solusinya.
- b. Pembuatan Model Sistem
Membuat pemodelan dalam UML agar lebih mudah dalam tahapan implementasi.

4 Implementasi dan Pembahasan

Tahapan ini meng-implementasikan perangkat yang telah dirancang. Apakah sudah bekerja sesuai dengan yang diharapkan. Adapun tahapan peng-implementasian yang dilakukan antara lain :

1. Menyalakan perangkat yang telah dibuat dan memastikan semua rangkaian telah bekerja dengan baik.
2. Memberikan sumber tegangan dari baterai 9 Volt.
3. Menekan tombol *push button* untuk memberikan nilai logika "1".
4. Melihat lampu indikator LED hidup atau mati.

IV. ANALISA DAN PERANCANGAN

Analisa sistem yang didapatkan dengan menggunakan metode *deskriptif* adalah perancangan rangkaian gerbang logika yang menggunakan baterai 9 V atau *power supply* sebagai *input* (masukan) tegangan, beberapa IC yang terdiri dari IC

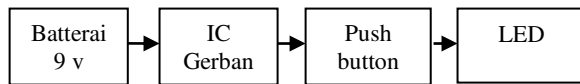
7408, 7432, 7400, 7402, 7404 sebagai gerbang logika dasar dengan terhubung pada *push button* untuk logika input 0 (Low) dan 1 (high) dan output keluaran yang dihasilkan adalah dengan lampu LED hidup untuk logika 1 (*High*) dan lampu LED mati untuk logika 0 (*Low*).

Gerbang logika dapat tersusun dari saklar sederhana, *relay*, transistor, dioda, atau IC. Gerbang logika dapat mempunyai beberapa masukan yang masing-masing mempunyai salah satu dari dua keadaan logika, yaitu 0 atau 1. Gerbang logika berdasarkan fungsinya dapat dibedakan menjadi beberapa jenis yaitu gerbang AND, OR, NAND, NOR, dan NOT.

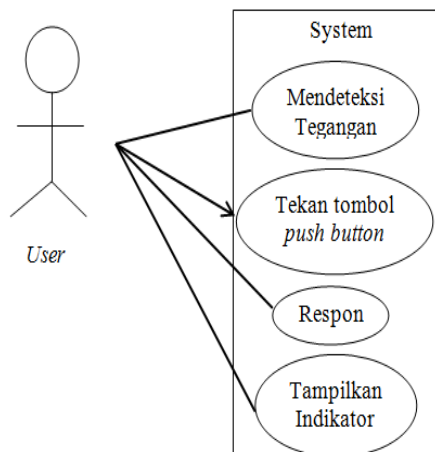
Ada beberapa operasi-operasi dasar pada suatu rangkaian logika dan untuk menunjukkan suatu perilaku dari operasi-operasi tersebut biasanya ditunjukkan dengan menggunakan suatu tabel kebenaran. Tabel kebenaran berisi statemen-statement yang hanya berisi:

1. Benar yang dilambangkan dengan huruf "T" kependekan dari "*True*" atau bisa jugadilambangkan dengan angka 1.
2. Salah yang dilambangkan dengan huruf "F" kependekan dari "*False*" atau bisa juga dilambangkan dengan angka 0.

Gerbang logika adalah piranti dua keadaan, yaitu mempunyai keluaran dua keadaan keluaran dengan nol yang menyatakan logika 0 (rendah) dan keluaran dengan tegangan tetap yang menyatakan logika 1 (tinggi). Rangkaian digital bipolar sebenarnya adalah rangkaian elektronika digital yang memanfaatkan transistor bipolar sebagai rangkaian gerbang.

A. *Arsitektur Sistem Rangkaian Gerbang Logika*

Gambar 4.1 Asitektur Sistem Rangkaian Gerbang Logika

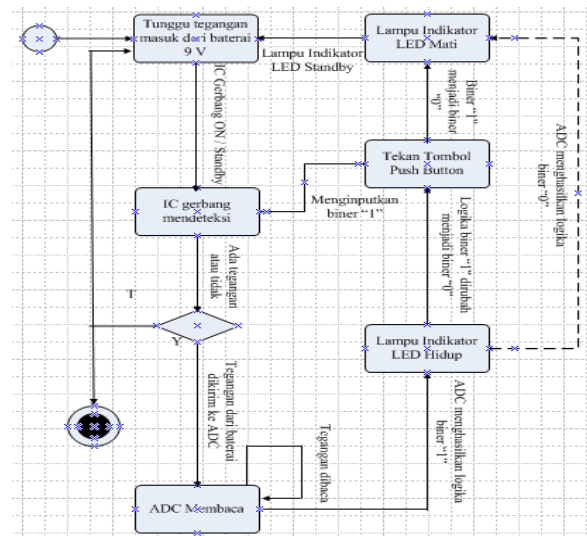
B. *Use Case Diagram*

Gambar 4.2 Use Case Diagram Sistem Gerbang Logika

User mendapatkan indikasi dari perangkat tentang keadaan yang terjadi. Indikasi dibuat dalam 4(Empat) keadaan. Bila IC gerbang menerima respon tegangan dan respon dari penekanan tombol *push button* maka respon akan ditampilkan pada lampu indikator sesuai dengan kondisi yang direspon oleh IC gerbang.

C. *State Machine Diagram*

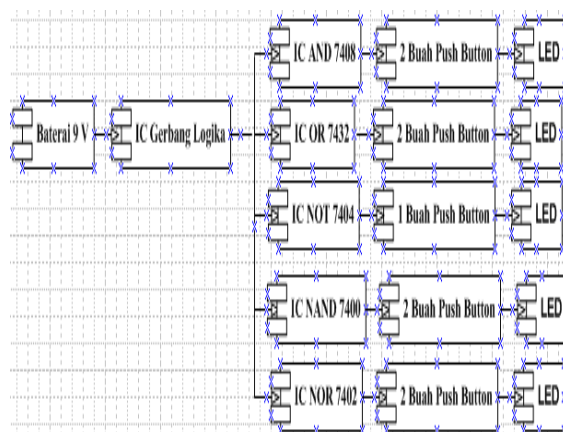
State Machine Diagram merupakan suatu penggambaran *behavior* sebuah sistem untuk sebuah *class* tunggal untuk menunjukkan *behavior* seumur hidup sebuah objek tunggal. Berikut ini gambar 4.3 *State Machine Diagram* aliran *behavior* pada sistem.



Gambar 4.3 State Machine Diagram Sistem Gerbang Logika Keseluruhan

D. *Component Diagram*

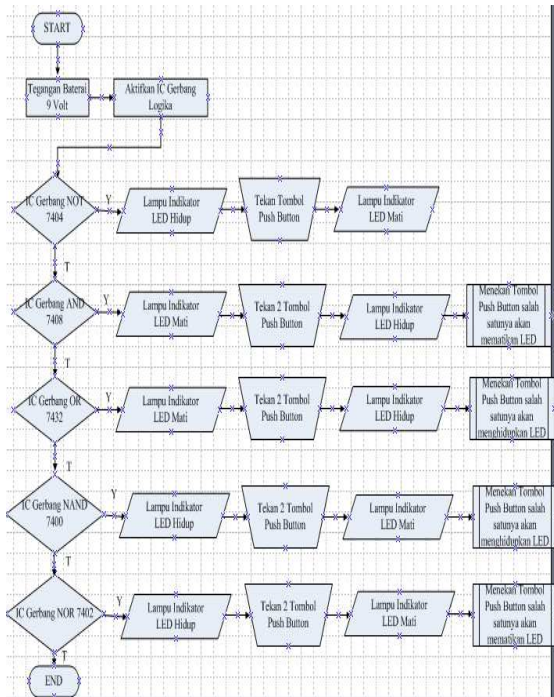
Component diagram merepresentasikan paket fisik dari modul pengkodean. *Component* diagram juga menunjukkan *interface* yang digunakan untuk komunikasi antar komponen. Berikut ini adalah gambar 4.4 *component* diagram dari sistem yang dibangun.



Gambar 4.4 Component Diagram Sistem Gerbang Logika

E. Diagram Alir / Flowchart Sistem

Di dalam mekanisme pembacaan kinerja sistem, maka penulis membuat dalam bentuk diagram alir / flowchart sistem seperti gambar 4.5



Gambar 4.5 Diagram Alir / Flowchart Sistem Gerbang Logika

V. IMPLEMENTASI DAN PEMBAHASAN

A. Tahapan Implementasi

Berikut ini tahapan implementasi sistem yang terdiri dari langkah-langkah sebagai berikut :

1. Menerapkan Rencana Implementasi

Dalam rencana implementasi ini, dimaksudkan untuk mengatur biaya dan waktu yang dibutuhkan selama implementasi sistem berjalan. Semua biaya yang akan dikeluarkan untuk kegiatan implementasi sistem ini perlu dianggarkan dalam bentuk anggaran biaya.

Karena anggaran biaya ini, berfungsi sebagai pengendalian terhadap biaya-biaya yang harus dikeluarkan. Waktu yang dibutuhkan untuk melakukan kegiatan juga diperlukan dalam rencana implementasi dalam bentuk *schedul* waktu. *Schedul* waktu juga berfungsi sebagai pengendalian terhadap waktu implementasi.

2. Menjalankan Kegiatan Implementasi

- Pemilihan dan pelatihan personil
- Pemilihan tempat dan instalasi perangkat keras dan perangkat lunak
- Pemrograman perangkat lunak dan Perancangan perangkat keras
- Pengujian sistem.

3. Tindak Lanjut Implementasi

Dalam hal ini setelah implementasi adalah pengujian. Pengujian disini dilakukan dengan cara mengujicoba perangkat yang telah selesai dirancang. Apakah telah sesuai dengan implementasi yang telah direncanakan sebelumnya.

B. Tujuan Implementasi

Tujuan implementasi adalah untuk mengetahui bagaimana sistem yang telah dirancang dapat berjalan sesuai dengan hasil rancangan. Dalam hal ini penulis merancang sebuah perangkat keras gerbang logika (OR, AND, NOT, NOR, dan NAND). Perangkat ini dirancang untuk memodelkan keadaan pemberian logika biner nol (0) atau satu (1) ke dalam IC gerbang logika. Agar mengetahui bagaimana sistem digital sebenarnya di dalam menghasilkan bilangan biner nol (0) atau satu (1) yang sering dijumpai pada sirkuit digital yang diimplementasikan secara elektronik.

C. Implementasi dan Pembahasan

Dari perancangan alat gerbang logika dasar yang telah dilakukan pada penelitian ini didapat hasil :

1. Perancangan Gerbang logika Awal



Gambar 5.1 Perancangan Awal Gerbang Logika dan Peletakkan Komponen

Perancangan awal gerbang logika dasar dengan meletakkan beberapa komponen diantaranya adalah IC gerbang, *push button* dan LED pada PCB dengan bantuan solder dan timah seperti yang ditunjukkan pada gambar 5.1 diatas.

Kemudian dihubungkan dengan batterai 9 V sebagai sumber tegangan seperti pada gambar 5.2 untuk menghidupkan *prototype* gerbang logika dasar.



Gambar 5.2 Perancangan Awal Gerbang Logika Akan Dihubungkan ke Baterai 9 Volt

2. Perakitan alat dengan sumber tegangan baterai 9 volt

Setelah alat dirakit, selanjutnya alat dihubungkan dengan sumber tegangan baterai 9 volt. Seperti pada gambar 5.3 dibawah ini :



Gambar 5.3 Perakitan Alat Dengan Sumber Tegangan Baterai 9 Volt

3. Hasil Perakitan dengan *Output* LED

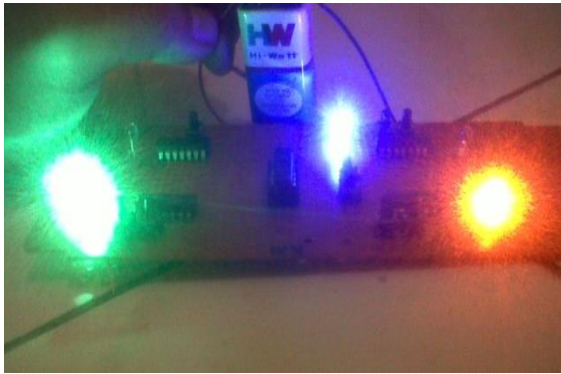
Prototype kemudian diuji coba dengan menekan *push button* untuk 2 input pada masing-masing gerbang diantaranya adalah gerbang AND, OR, NOT, NAND dan NOR seperti pada gambar 5.4.



Gambar 5.4 Hasil Perakitan Dengan *Output* LED

4. Hasil Perakitan Dengan Logika Inputan *Push Button*

Pada hasil perakitan ini dilakukan pengujian yang ditunjukkan pada gambar 5.5 dengan peng-inputan logika biner satu (1), dengan menekan tombol *push button* pada masing-masing IC gerbang logika yang outputnya adalah lampu indikator LED hidup yang dapat dilihat langsung pada rangkaian gerbang logika.



Gambar 5.5 Hasil Perakitan Dengan Logika Inputan
Push Button

VI. KESIMPULAN DAN SARAN

A. Kesimpulan

Dari hasil perancangan gerbang logika dasar dapat disimpulkan :

1. Penelitian *prototype* gerbang logika sangat membantu sekali bagi mahasiswa dalam materi kuliah rangkaian digital dan sistem digital untuk pembuktian logika dasar
2. Gerbang logika dasar adalah gerbang yang memiliki 2 (dua) *input* (masukan) dan 1 (satu) *output* (keluaran)
3. Gerbang logika dasar terdiri dari output (keluaran) yaitu 0 (*Low*) dan 1 (*High*).

B. Saran

Sebagai saran dari penelitian ini, adalah :

1. Penelitian *prototype* gerbang logika masih belum sempurna, agar dilanjutkan penelitian selanjutnya dalam keluaran yang menggunakan angka
2. *Prototype* gerbang logika ini dapat dikembangkan lagi ke depannya dengan menambahkan gerbang-gerbang logika lainnya seperti gerbang XOR, dan XNOR.

DAFTAR PUSTAKA

- [1] Abdul Jabbar (2011). "Pemodelan dan Simulasi Dinamis Pendeteksi Dini Gempa Pada Gedung." Universitas Putra Indonesia YPTK Padang : Tesis M.KOM.
- [2] Albert, Paul & Tjia. 1994. *Elektronika Komputer Digital & Pengantar Komputer* Edisi 2. Jakarta : Erlangga.
- [3] Ibrahim, KF, Teknik Digital, Andi Offset, Yogyakarta, 1996
- [4] Kasmawan, Antha. 2010. *Penuntun Praktikum Elektronika 2*. Jimbaran : Unud.
- [5] Kurniawan, Fredly. 2005. *Sistem Digital Konsep & Aplikasi*. Yogyakarta : Gava Media.
- [6] Malvino dkk., Prinsip prinsip penerapan digital, Penerbit Erlangga, Surabaya, edisi ketiga
- [7] Martin Fowler. (2005). "UML Distilled." 3th. Ed. A Brief Guide to the Standard Object Modelling Language : Andi. 151-160.
- [8] Mowle, J, Frederic, A systematic Approach to Digital Logic Design, Addison Wesley, 1976
- [9] Muhsin. 2004. *Elektronika Digital Teori & Soal Penyelesaian*. Yogyakarta : Graha Ilmu.
- [10] Uffenbeck, John, Microcomputer and Microprosesor, Second edition, Prentice Hall International, Inc, 1985